(or the master-client) then sends an email to those users listed by the master user. These emails contain a 'n-client application' (herein n-client) as an attachment. When user n_j executes the n-client, it performs as the client application above. (Note: in this case the n-client n_j for user checks under the n_j entry within the master user's information file stored at the server.)

If user n_j is authenticated, the n-client downloads the n-loader. In this case, the n-loader may not contain the protected application. Should the n-loader's quorum be reached, among the usually tasks performed, it sends a message to the master user/vendor/or other server site, that user n_j is ready for the protected application which is then sent via a compact or floppy disk or by other physical means.

Alternatively, the protected application may be included within the n-loader (analogous to above). In this case it will have been downloaded to the n_j user.

When the protected application is executed, it invokes the linked library of the invention as described above. Should the quorum be reached, the n_j user is allowed access into the protected application as described above.

Alternatively, the master-client application may send, rather than an e-mails as mentioned above, a URL to $n_1, n_2,\ldots,n_j,\ldots n_n$. In this case, the URL will allow users to go to a special portion of the server to download the n-client. This would allow the master-user to pre-purchase/pre-license a multiple user licenses. Then, at the download area of the server, only a pre-set number of n-clients will be available to be downloaded.

Although the invention has been described with respect to various embodiments, it should be realized this invention is also capable of a wide variety of further and other embodiments within the spirit and scope of the appended claims.

What is claimed is

1. A process for protecting software products from unauthorized usage comprising the steps of:

determining a set of parameters associated with the user's system,

sending the parameters to a server,

forming a first value derived from the parameters,

sending the first value to the user's system,

determining a second set of the same parameters from the user's system,

using the second set of parameters to form a second version of the first value,

comparing the first and the second values, and if identical,

allowing the software product to be executed in the user's system.

2. The process as defined in claim 1 further comprising the steps of:

encrypting the software product using the first value, and

if the first and the second values are identical, sending the software product to the user's system.

3. A process for protecting software products from unauthorized usage, comprising the steps of:

determining a set of parameters associated with the user's system,

sending the parameters to a server,

forming a first value derived from the parameters,

sending the first value to the user's system,

determining a second set of the same parameters from the user's system,

using the second set to form a second version of the first value,

comparing the first and the second values, and if identical,

sending the software product to the user's system.

4. The process as defined in claim 3 further comprising the step of: encrypting the software product using the first value.

5. The process as defined in claim 4, further comprising the step of: if the first value is identical to the second value, decrypting the software product using the value.

6. The process as defined in claim 1 further comprising the step of: encrypting the software product using the first value.

7. The process as defined in claim 6, further comprising the step of: decrypting the software product using the second value if the first value is identical to the second value.

8. The process as defined in claim 1 or 3, wherein the first value is encrypted by the parameters, and later decrypted to form the second version of the first value.

9. The process as defined in claim 1 or 3, wherein the first value is encrypted for each member of the set of parameters, and later decrypted to form the second version of the first value.

10. The process as defined in claims 1 or 3, wherein the first value is encrypted for each member of the set of parameters, and later decrypted by using each member of the new set of parameters to form a set of second versions of the first value.

11. The process as defined in claims 9 ~~or 10~~ further comprising the step of: SC 6/8/00

determining if a quorum of identical members of the set of the second version of the first value exists.

12. The process as defined in claim 11, wherein if the number of identical members satiisfy the quorum condition, then the user's system is allowed to receive and execute the software product.

13. A system for protecting software products from unauthorized usage, comprising:

parameters associated with the user's system,

means for sending the parameters to a server,

a first value derived from the parameters,

means for sending the first value to the user's system,

a second set of the same parameters associated with the user's system,

a second value derived from the second set of parameters in the identical fashion as the first value was derived,

a comparator that compares the first and the second values, and if identical,

means for allowing the software product to be executed in the user's system.

14. The system as defined in claim 13 further comprising:

means for encrypting the software product using the first value, and if allowed

means for delivering the software product to the user's system.

15. A system for protecting software products from unauthorized usage, comprising:

parameters associated with a user's client computing system,

a server,

means for sending the parameters to the server,

means for the server to formulate a first value from the set of received parameters,

means for encrypting the value using each parameter as a key thereby forming a set of encrypted values,

means for encrypting the software product,

means for downloading the set of encrypted values, and the encrypted software product to the client,

means for determining new values of the same set of parameters associated with the user's system

means for decrypting each encrypted value using a member of the new set of parameters,

means for comparing the set of values and the first value to determine if the user should be authorized, if authorized the server enables the client to decrypt and run the software product.